Digital image processing techniques for the detection and removal of cracks in digitized paintings

Ioannis Giakoumis, Nikos Nikolaidis, Ioannis Pitas
Department of Informatics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
tel/fax: +302310996304
e-mail: {*nikolaid,pitas*}*@zeus.csd.auth.gr*

Abstract

An integrated methodology for the detection and removal of cracks on digitized paintings is presented in this paper. The cracks are detected by thresholding the output of the morphological top-hat transform. Afterwards, the thin dark brush strokes which have been misidentified as cracks are removed using either a Median Radial Basis Function (MRBF) neural network on hue and saturation data or a semi-automatic procedure based on region growing. Finally, crack filling using order statistics filters or controlled anisotropic diffusion is performed. The methodology has been shown to perform very well on digitized paintings suffering from cracks.

I. INTRODUCTION

Many paintings, especially old ones, suffer from breaks in the substrate, the paint, or the varnish. These patterns are usually called cracks or craquelure and can be caused by aging, drying, and mechanical factors. Age cracks can result from non-uniform contraction in the canvas or wood-panel support of the painting, which stresses the layers of the painting. Drying cracks are usually caused by the evaporation of volatile paint components and the consequent shrinkage of the paint. Finally, mechanical cracks result from painting deformations due to external causes, e.g. vibrations and impacts.

The appearance of cracks on paintings deteriorates the perceived image quality. However, one can use digital image processing techniques to detect and eliminate the cracks on digitized paintings. Such a ”virtual” restoration can provide clues to art historians, museum curators and the general public on how the painting would look like in its initial state, i.e., without the cracks. Furthermore, it can be used as a non-destructive tool for the planning of the actual restoration. A system that is capable of tracking and interpolating cracks is presented in [1]. The user should manually select a point on each crack to be restored. A method for the detection of cracks using multi-oriented

Gabor filters is presented in [2]. Crack detection and removal bears certain similarities with methods proposed for the detection and removal of scratches and other artifacts from motion picture films [3], [4], [5]. However, such methods rely on information obtained over several adjacent frames for both artifact detection and filling and thus are not directly applicable in the case of painting cracks. Other research areas that are closely related to crack removal include image inpainting which deals with the reconstruction of missing or damaged image areas by filling-in information from the neighboring areas, and disocclusion, i.e., recovery of object parts that are hidden behind other objects within an image. Methods developed in these areas assume that the regions where information has to be filled-in are known. Different approaches for interpolating information in structured [6], [7], [8], [9], [10] and textured image areas [11] have been developed. The former are usually based on partial differential equations (PDE) and on the calculus of variations whereas the latter rely on texture synthesis principles. A technique that decomposes the image to textured and structured areas and uses appropriate interpolation techniques depending on the area where the missing information lies has also been proposed [12]. The results obtained by these techniques are very good. A methodology for the restoration of cracks on digitized paintings, which adapts and integrates a number of image processing and analysis tools is proposed in this paper. The methodology is an extension of the crack removal framework presented in [13]. The technique consists of the following stages:

- Crack detection.
- Separation of the thin dark brush strokes, which have been misidentified as cracks.
- Crack filling (interpolation).

A certain degree of user interaction, most notably in the crack detection stage, is required for optimal results. User interaction is rather unavoidable since the large variations observed in the typology of cracks would lead any fully automatic algorithm to failure. However, all processing steps can be executed in real time and thus the user can instantly observe the effect of parameter tuning on the image under study and select in an intuitive way the values that achieve the optimal visual result. Needless to say that only subjective optimality criteria can be used in this case since no ground truth data are available. The opinion of restoration experts that inspected the virtually restored images was very positive.

This paper is organized as follows. Section II describes the crack detection procedure. Two methods for the separation of the brush strokes which have been falsely identified as cracks are presented in Section III. Methods for filling the cracks with image content from neighboring pixels are proposed in Section IV. Conclusions and discussion follow.

II. Detection of cracks

Cracks usually have low luminance and thus can be considered as local intensity minima with rather elongated structural characteristics. Therefore, a crack detector can be applied on the luminance component of an image and should be able to identify such minima. A crack detection procedure based on the so-called top-hat transform [14] is proposed in this paper. The top-hat transform is a grayscale morphological filter defined as follows:

$$y(x) = f(x) - f_{nB}(x) \tag{1}$$

where $f_{nB}(x)$ is the opening of the function $f(x)$ (in our case, the luminance component of the image under study) with the structuring set nB, defined as:

$$nB = B \oplus B \oplus \cdots \oplus B \qquad (n\ times) \tag{2}$$

In the previous equation $\oplus$ denotes the dilation operation. A square or a circle can be used as structuring element B [15]. The final structuring set nB is evaluated only once using (2) and is used subsequently in the opening operation of (1). The opening f_{nB} of a function is a low-pass nonlinear filter that erases all peaks (local maxima) in which the structuring element nB cannot fit. Thus, the image $f - f_{nB}$ contains only those peaks and no background at all. Since cracks are local minima rather than local maxima the top-hat transform should be applied on the negated luminance image. Alternatively, one can detect cracks by performing closing on the original image $f(x)$ with the structuring set nB and then subtracting $f(x)$ from the result of closing $f^{nB}(x)$:

$$y(x) = f^{nB}(x) - f(x) \tag{3}$$

It can be easily shown that the result of (3) is identical to that of applying (1) on the negated image. Use of (3) does not require negation of $f(x)$ which grands it a small but not negligible computational advantage over (1).

In situations where the crack-like artifacts are of high luminance, as in the case of scratches on photographs, negation of the luminance component prior to the crack detection is not required, i.e. the crack detection procedure can be applied directly on the luminance image. The user can control the result of the crack detection procedure by choosing appropriate values for the following parameters:

- The type of the structuring element B.
- The size of the structuring element B and the number n of dilations in (2).

These parameters affect the size of the ”final” structuring element nB and must be chosen according to the thickness of the cracks to be detected. It should be noted however that these parameters are not very critical for the algorithm performance due to the thresholding operation that will be described in the next paragraph and also due to the existence of the brush stroke / crack separation procedure (section III), which is able to remove crack-like brush strokes that have been erroneously identified as cracks. The fact that all the results presented in this paper have been obtained with the same top-hat transform parameters comes as a clear indication that the above statement is indeed true. These parameters were the following:

- Structuring element type: square
- Structuring element size: 3×3
- Number n of dilations in (2): 2

The top-hat transform generates a grayscale output image $t(k, l)$ where pixels with a large grey value are potential crack or crack-like elements. Therefore, a thresholding operation on $t(k, l)$ is required to separate cracks from the rest of the image. The threshold T can be chosen by a trial and error procedure, i.e., by inspecting its effect on the resulting crack map. The low computational complexity of the thresholding operation enables the user to view the crack detection results in real time while changing the threshold value. e.g., by moving a slider. This fact makes interactive threshold selection very effective and intuitive. Alternatively, threshold selection can be done by inspecting the histogram of $t(k, l)$ for a lobe close to the maximum intensity value (which will most probably correspond to crack or crack-like pixels), and assigning it a value that separates this lobe from the rest of the intensities. The result of the thresholding is a binary image $b(k, l)$ marking the possible crack locations. Instead of this global thersholding technique, more complex thresholding schemes, which use a spatially varying threshold can be used. Obviously, as the threshold value increases the number of image pixels that are identified as cracks decreases. Thus, certain cracks, especially in dark image areas where the local minimum condition may not be satisfied, can remain undetected. In principle, it is more preferable to select the threshold so that some cracks remain undetected than to choose a threshold that would result in the detection of all cracks but will also falsely identify as cracks, and subsequently modify, other image structures. The thresholded (binary) output of the top-hat transform on the luminance component of an image containing cracks (Figure 1) can be seen in Figure 2. Additional examples of cracks detected using this approach can be seen in Figures 10-12.

III. Separation of the brush strokes from the cracks

In some paintings, certain areas exist where brush strokes have almost the same thickness and luminance features as cracks. The hair of a person in a portrait could be such an area. Therefore, the top-hat transform might misclassify these dark brush strokes as cracks. Thus, in order to avoid any undesirable alterations to the original image, it is very important to separate these brush strokes from the actual cracks, before the implementation of the crack filling procedure. Two methods to achieve this goal are described in the following subsections.

A. Semi-automatic crack separation

A simple interactive approach for the separation of cracks from brush strokes is to apply a region growing algorithm on the thresholded output of the top-hat transform, starting from pixels (seeds) on the actual cracks. The pixels are chosen by the user in an interactive mode. At least one seed per connected crack element should be chosen. Alternatively, the user can choose to apply the technique on the brush strokes, if this is more convenient. The growth mechanism that was used implements the well-known grassfire algorithm that checks recursively for unclassified pixels with value 1 in the 8-neighborhood of each crack pixel. At the end of this procedure, the pixels in the binary image, which correspond to brush strokes that are not 8-connected to cracks will be removed. The above procedure can be used either in a stand-alone mode or applied on the output of the MRBF separation procedure described in the next section to eliminate any remaining brush strokes.

B. Discrimination on the basis of hue and saturation

Hue H is associated with the dominant wavelength in a mixture of light wavelengths and represents the dominant color. In the HSV color model, hue is represented as the angle around the vertical axis, with red at $0°$, green at $120°$, and so on. Saturation S refers to the amount of white light mixed with a certain hue. Hue and saturation are defined similarly in other related color domains, e.g. in Hue Saturation Intensity (HSI) or Hue Lightness Saturation (HLS).

By statistical analysis of 47 digitized paintings (24 portable religious icons from the Byzantine era and 23 paintings of various styles and ages), it has been concluded that the hue of the cracks usually ranges from $0°$ to $60°$. On the contrary, we observed that the hue of the dark brush strokes varies, as expected, in the entire gamut [$0°$, $360°$]. Furthermore, crack saturation usually ranges from 0.3 to 0.7, while brush stroke saturation ranges from 0 to 0.4. Thus, on the basis of these observations, a great portion of the dark brush strokes, falsely detected by the top-hat transform, can be separated from the cracks. This separation can be achieved by classification using a Median Radial Basis Function (MRBF) neural network, which is a robust, order statistics based, variation of Radial Basis Function (RBF) networks [16].

RBFs are two-layer feedforward neural networks [17], that model a mapping between a set of input vectors and a set of outputs. The network architecture is presented n Figure 3. RBFs incorporate an intermediate, hidden layer where each hidden unit implements a kernel function, usually a Gaussian function:

$$\phi_j(\boldsymbol{X}) = \exp[(\boldsymbol{\mu}_j - \mathbf{X})^T \boldsymbol{S}_j^{-1} (\boldsymbol{\mu}_j - \mathbf{X})], \;\; j = 1, \ldots, L \tag{4}$$

where $\boldsymbol{\mu}_j$, $\boldsymbol{S}_j$ denote the mean vector and the covariance matrix for kernel j and L denotes the number of units (kernels) in the hidden layer. Each output consists of a weighted sum of kernels. In typical situations that involve pattern classification, the number of outputs equals to the number of classes. In such a setting, the current vector is assigned to the class associated with the output unit exhibiting the maximum activation (winner takes all approach). After the learning stage, the network implements the input-output mapping rule and can generalize it to input vectors not being part of the training set.

The parameters to be estimated (learned) in a RBF network are the center (mean) vector $\boldsymbol{\mu}_j$ and the covariance matrix $\boldsymbol{S}_j$ for each Gaussian function and the weights $w_{k,j}$ corresponding to the connections between neurons in the hidden layer and output nodes. A hybrid technique that has been frequently used for the training of such networks, has been adopted for the learning stage. According to this technique, training is performed in two successive steps: the hidden layer parameters are estimated using an unsupervised approach and, afterwards, the output layer weights are updated in a supervised manner, using the (now fixed) hidden layer parameters evaluated in the previous step.

In the classical version of the adopted training technique, a variation of the Learning Vector Quantizer (LVQ) algorithm is used for the unsupervised hidden layer parameter updating. Each input vector is assigned to the Gaussian kernel whose center is closer (in terms of either the Euclidean or the Mahalanobis distance) to this vector:

$$if \parallel \mathbf{X}_i - \boldsymbol{\mu}_j \parallel = min_{k=1}^{L} \parallel \mathbf{X}_i - \boldsymbol{\mu}_k \parallel \;\; then \, \mathbf{X}_i \in C_j \tag{5}$$

where $\parallel \, \parallel$ denotes either Euclidean or Mahalanobis distance and C_j denotes the class of input vectors associated with kernel ϕ_j. Subsequently, the algorithm updates the center and covariance matrix of the winner kernel using running versions of the classical sample mean and sample covariance matrix formulas. On the other hand, the MRBF algorithm which has been used in our case is based on robust estimation [16] of the hidden unit parameters.

It employs the Marginal Median LVQ [18] that selects the winner kernel using (5) and utilizes the marginal median of the input vectors currently assigned to this kernel for the update of the center vector (location parameter) $\boldsymbol{\mu}_j$ of the kernel:

$$\boldsymbol{\mu}_j = marginal\_median\{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_{n-1}\}; \tag{6}$$

where $\mathbf{X}_{n-1}$ is the last vector assigned to kernel j. The update of the diagonal elements of the corresponding covariance matrix is performed using the median of the absolute deviations (MAD) [19] of the inputs currently assigned to this kernel:

$$\boldsymbol{\sigma}_j = \frac{marginal\_median\{|\mathbf{X}_0 - \boldsymbol{\mu}_j|, \ldots, |\mathbf{X}_{n-1} - \boldsymbol{\mu}_j|\}}{0.6745} \tag{7}$$

In the previous expression, $\boldsymbol{\sigma}_j$ denotes the vector containing the diagonal elements of the covariance matrix and $|\mathbf{X}|$ denotes the vector obtained by taking the absolute value of each component of $\mathbf{X}$. The off-diagonal components of the covariance matrix are also calculated based on robust statistics [16]. In order to avoid excessive computations the above operations can be applied on a subset of data extracted through a moving window that contains only the last W data samples assigned to the hidden unit j.

In the supervised part of the learning procedure, the weights of the output layer, which group the clusters found by the hidden layer into classes, are updated. The update mechanism for these weights is described by the following expression:

$$w_{k,j}(t+1) = w_{k,j}(t) + n_w(F_k(\mathbf{X}) - Y_k(\mathbf{X}))Y_k(\mathbf{X})(1 - Y_k(\mathbf{X}))\phi_j(\mathbf{X}) \tag{8}$$

for $k = 1, \ldots, M$, $j = 1, \ldots, L$, and a learning factor $n_w \in (0, 1]$. In the previous formula $Y_k(\mathbf{X})$, $F_k(\mathbf{X})$ denote the actual and desired network output for input vector $\mathbf{X}$. The latter is given by:

$$F_k(X) = \begin{cases} 1 & if\ \mathbf{X} \in C_k \\ 0 & if\ \mathbf{X} \notin C_k \end{cases} \tag{9}$$

The update formula (8) corresponds to the backpropagation for the output weights of a RBF network with respect to the mean square error cost function.

In our implementation, a MRBF network with two outputs was used. The first output represents the class of cracks while the second one the class of brush strokes. Input vectors were two-dimensional and consisted of the hue and saturation values of pixels identified as cracks by the top-hat transform. The number of clusters (hidden units) chosen for each class depends on the overlap between the populations of cracks and brush strokes. If there is a substantial overlap, the number should be increased, in order to reduce the classification error. In our implementation three hidden units have been incorporated. Training was carried out by presenting the network with hue and saturation values for pixels corresponding to cracks and crack-like brush strokes. Data from 24 digitized portable religious icons from the Byzantine era were used for this purpose. The system trained using this specific training set can be considered to be optimized for paintings of this style and its use on paintings of other style might result in somewhat suboptimal results. However, appropriately selected training sets can be used to train the system to separate cracks from brush strokes on paintings of different artistic styles or content. In order to select pixels corresponding to cracks and crack-like brush strokes the crack detection algorithm presented in section II was applied on these images. Results were subsequently post-processed by an expert using the semi-automatic approach presented in section III-A. The aim of this post-processing step was twofold: to remove pixels that are neither cracks nor crack-like brush strokes and to separate cracks and crack-like brush strokes for the supervised step of the training procedure. In this supervised training step, the network was presented with these labelled inputs, i.e., pairs of hue-saturation values that corresponded to image pixels that have been identified as belonging to cracks and crack-like brush strokes.

After the training session, the MRBF neural network was able to classify pixels identified as cracks by the top hat transform to cracks or brush strokes. The trained network has been tested on 12 images from the training set and 15 images (of the same artistic style and era) that did not belong to the training set. Naturally, the performance of the cracks / brush strokes separation procedure was judged only in a subjective manner (i.e. by visual inspection of the results), as ground truth data (i.e. brush strokes-free crack images) are not available. For this reason two restoration experts were asked to inspect several crack images before and after the application of the separation system and concluded that in the processed crack images the great majority of the brush strokes has been removed. A thresholded top-hat transform output containing many brush strokes e.g. hair is illustrated in Figure 5. A great part of these brush strokes is separated by the MRBF, as can be seen in Figure 6. The original image can be seen in Figure 4.

IV. Crack filling methods

After identifying cracks and separating misclassified brush strokes, the final task is to restore the image using local image information (i.e., information from neighboring pixels) to fill (interpolate) the cracks. Two classes of techniques, utilizing order statistics filtering and anisotropic diffusion are proposed for this purpose. Both are implemented on each RGB channel independently and affect only those pixels which belong to cracks. Therefore, provided that the identified crack pixels are indeed crack pixels, the filling procedure does not affect the ”useful” content of the image. Image inpainting techniques like the ones cited in Section I can also be used for crack filling.

The performance of the crack filling methods presented below was judged by visual inspection of the results. Obviously, measuring the performance of these methods in an objective way is infeasible since ground truth data (e.g. images depicting the paintings in perfect condition, i.e., without cracks) are not available. For the evaluation of the results, two restoration experts were asked to inspect several images restored using the various methods and comment, based on their experience, on the quality of the filling results, (i.e., whether cracks were sufficiently filled), whether the color used for filling was the correct one, whether fine image details were retained, etc.

A. Crack filling based on order statistics filters

An effective way to interpolate the cracks is to apply median or other order statistics filters [15] in their neighborhood. All filters are selectively applied on the cracks, i.e., the center of the filter window traverses only the crack pixels. If the filter window is sufficiently large, the crack pixels within the window will be outliers and will be rejected. Thus, the crack pixel will be assigned the value of one of the neighboring non-crack pixels. The following filters can be used for this purpose:

- Median filter:

$$y_i = med(x_{i-\nu}, \ldots, x_i, \ldots, x_{i+\nu}) \tag{10}$$

- Recursive median filter:

$$y_i = med(y_{i-\nu}, \ldots, y_{i-1}, x_i, \ldots, x_{i+\nu}) \tag{11}$$

  where the $y_{i-\nu}, \ldots, y_{i-1}$ are the already computed median output samples. For both the recursive median and the median filter, the filter window (considering only rectangular windows) should be approximately 50$\%$ wider than the widest (thickest) crack appearing on the image. This is necessary to guarantee that the filter output is selected to be the value of a non-crack pixel. Smaller windows will result in cracks that will not be sufficiently filled whereas windows that are much wider than the cracks will create large homogeneous areas, thus distorting fine image details.

- Weighted median filter:

$$y_i = med(w_{-\nu}\Diamond x_{i-\nu}, \ldots, w_{\nu}\Diamond x_{i+\nu}) \tag{12}$$

  where $w\Diamond x$ denotes duplication of x, w times. For this filter, smaller filter windows (e.g. windows that are approximately 30$\%$ wider than the widest crack appearing on the image) can be used since the probability that a color value corresponding to a crack is selected as the filter output (a fact that would result in the crack pixel under investigation not being filled effectively by the filter) can be limited by using small weights (e.g. 1) for the pixels centrally located within the window (which are usually part of the crack) and bigger ones (e.g. 2 or 3) for the other pixels.

- A variation of the modified trimmed mean (MTM) filter which excludes the samples $x_{i+r,j+s}$ in the filter window, which are considerably smaller from the local median and averages the remaining pixels:

$$y_{ij} = \frac{\sum\sum_A \alpha_{rs} x_{i+r,j+s}}{\sum\sum_A \alpha_{rs}} \tag{13}$$

  The summations cover the entire filter window A. The filter coefficients are chosen as follows:

$$\alpha_{rs} = \begin{cases} 0 & \text{if } med\{x_{ij}\} - x_{i+r,j+s} \;\geq\; q \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

  The amount of trimming depends on the positive parameter q. Data of small value deviating strongly from the local median (which correspond usually to cracks) are trimmed out. Windows used along with this variant of the MTM filter can also be smaller than those used for the median and recursive median filters since a portion of the crack pixels is expected to be rejected by the trimming procedure.

- Another variation of the MTM filter that performs averaging only on those pixels that are not part of the crack, i.e., it utilizes information from the binary output image $b(k, l)$ of the top hat transform. In this case, the filter coefficients in (13) are chosen as follows:

$$\alpha_{rs} = \begin{cases} 1 & \text{if } b(k,l) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The median operator can be used instead of the arithmetic mean in (13). For this variant of the MTM filter, even smaller filter windows can be used, since crack pixels do not contribute to the filter output. Thus, it suffices that the window is 1 pixel wider than the widest crack.

The result of the application of the second variation of the modified trimmed mean filter on the painting depicted in Figure 1a is shown in Figure 7 (filter size 5×5). Another image restored by the same crack-filling approach can be seen in Figure 8 (filter size 3×3). Extensive experimentation proved that this filter gives the best results among all filters presented above according to subjective evaluation by restoration experts. The superiority of this filter can be attributed to the fact that only non-crack pixels contribute to its output. On the contrary, for all other filters presented in this section the probability that crack pixels will contribute to the output is small but not negligible.

B. Controlled Anisotropic Diffusion

Anisotropic diffusion [20] is an image enhancement method that successfully combines smoothing of slowly varying intensity regions and edge enhancement. Smoothing is modelled as a diffusion that is allowed along homogeneous regions and inhibited by region boundaries. Anisotropic diffusion is described by the following partial differential equation:

$$\frac{\partial I(x,y,t)}{\partial t} = div(c(x,y,t)\nabla I(x,y,t)) = c(x,y,t)\Delta I(x,y,t) + \nabla c(x,y,t) \cdot \nabla I(x,y,t) \tag{16}$$

where *div* denotes the divergence operator and ∇, Δ the gradient and Laplacian operators with respect to the space variables x, y. At each position and iteration, diffusion is controlled by the conduction (or diffusion) coefficients $c(x,y,t)$. Since diffusion should be inhibited across regions separated by discontinuities, the conduction coefficients should obtain small values in pixels with large intensity gradient magnitude. A similar approach named Curvature Driven Diffusion (CCD) has been proposed in [9] for image inpainting applications.

In order to obtain a numerical solution to the diffusion equation, discretization of the spatial and time coordinates and approximation of the differential operators by finite difference operators should be performed in (16). We have used the same discretizations as proposed in [20]. The iterative, discrete solution to (16) is governed by the following equation:

$$I_{i,j}^{t+1} = I_{i,j}^{t} + \lambda[c_N \cdot D_N I + c_S \cdot D_S I + c_E \cdot D_E I + c_W \cdot D_W I]_{i,j}^{t} \tag{17}$$

where $0 \leq \lambda \leq 1/4$ for the scheme to be stable, N, S, E, W are the mnemonics for North, South, East, West and the symbol D indicates nearest-neighbor differences:

$$\begin{aligned} D_N I_{i,j} &\equiv I_{i-1,j} - I_{i,j} \\ D_S I_{i,j} &\equiv I_{i+1,j} - I_{i,j} \\ D_E I_{i,j} &\equiv I_{i,j+1} - I_{i,j} \\ D_W I_{i,j} &\equiv I_{i,j-1} - I_{i,j} \end{aligned} \tag{18}$$

The conduction coefficients are evaluated at every iteration as a function $g(\cdot)$ of the magnitude of the intensity gradient. In our implementation, the following approximation was used [20]:

$$\begin{aligned} c^t_{N_{i,j}} &= g\ (\mid\ D_N I^t_{i,j}\ \mid) \\ c^t_{S_{i,j}} &= g\ (\mid\ D_S I^t_{i,j}\ \mid) \\ c^t_{E_{i,j}} &= g\ (\mid\ D_E I^t_{i,j}\ \mid) \\ c^t_{W_{i,j}} &= g\ (\mid\ D_W I^t_{i,j}\ \mid) \end{aligned} \tag{19}$$

The following function $g(\cdot)$, proposed in [20], has been used in our case:

$$g(\nabla I) = \frac{1}{1 + (\frac{\|\nabla I\|}{K})^2} \tag{20}$$

The constant K was manually fixed. In order to fill the cracks, the anisotropic diffusion algorithm was applied selectively, in neighborhoods centered on crack pixels. All pixels within these neighborhoods participate in the diffusion process. However, only the values of the crack pixels are updated in the output image.

Further improvements were obtained by taking into account crack orientation, i.e., by applying the operation only in a direction perpendicular to the crack direction. For example, if the crack is horizontal, one can use only the North and the South neighbors, since the West and the East neighbors belong also to the crack. In order to find the directions of the cracks, the Hough Transform was applied [21]. The restoration results of this filter can be seen in Figures 9-12. The orientation-sensitive controlled anisotropic diffusion method gave the best results among all crack-filling methods presented in this paper. The fact that this filter requires no window size selection grants it an additional advantage over the order statistics filters presented in the previous section.

V. Conclusions and Discussion

In this paper, we have presented an integrated strategy for crack detection and filling in digitized paintings. Cracks are detected by using top-hat transform, whereas the thin dark brush strokes, which are misidentified as cracks, are separated either by an automatic technique (MRBF networks) or by a semi-automatic approach. Crack interpolation is performed by appropriately modified order statistics filters or controlled anisotropic diffusion. The methodology has been applied for the virtual restoration of images and was found very effective by restoration experts. However, there are certain aspects of the proposed methodology that can be further improved. For example, the crack detection stage is not very efficient in detecting cracks located on very dark image areas, since in these areas the intensity of crack pixels is very close to the intensity of the surrounding region. A possible solution to this shortcoming would be to apply the crack detection algorithm locally on this area and select a low threshold value. Another situation where the system (more particularly, the crack filling stage) does not perform as efficiently as expected is in the case of cracks that cross the border between regions of different color. In such situations, it might be the case that part of the crack in one area is filled with color from the other area, resulting in small spurs of color in the border between the two regions. Such a situation is depicted in Figure 11. However this phenomenon is rather seldom and, furthermore, the extent of these erroneously filled areas is very small (2-3 pixels maximum). A possible solution would be to perform edge detection or segmentation on the image and confine the filling of cracks that cross edges or region borders to pixels from the corresponding region. Use of image inpainting techniques [6], [7], [8], [9], [10] could also improve results in that aspect. Another improvement of the crack filling stage could aim at using properly adapted versions of nonlinear multichannel filters (e.g. variants of the vector median filter) instead of processing each color channel independently. These improvements will be the topic of future work on this subject.

REFERENCES

[1] M. Barni, F. Bartolini, V. Cappellini, ”Image processing for virtual restoration of artworks”, *IEEE Multimedia*, vol. 7, no. 2, pp. 34-37, April-June 2000.

[2] F. Abas, K. Martinez, ”Craquelure Analysis for Content-based Retrieval”, *in Proc. 14th International Conference on Digital Signal Processing*, 2002, vol 1, pp. 111-114.

[3] L. Joyeux, O. Buisson, B. Besserer, S. Boukir, ”Detection and removal of line scratches in motion picture films,” *in Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 1999, pp. 548-553.

[4] A. Kokaram, R. Morris, W. Fitzgerald, P. Rayner, ”Detection of missing data in image sequences”,*IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1496-1508, November 1995.

[5] A. Kokaram, R. Morris, W. Fitzgerald, P. Rayner, ”Interpolation of missing data in image sequences”, *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1509-1519, November 1995.

[6] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, ”Image Inpainting”, *in Proc. SIGGRAPH 2000*, pp. 417–424, 2000.

[7] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, ”Filling-In by Joint Interpolation of Vector Fields and Gray Levels”, *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, August 2001.

[8] S. Masnou, J.M. Morel, ”Level Lines Based disocclusion”, *in Proc. IEEE ICIP'98*, vol. III, pp. 259–263, 1998.

[9] T. Chan, J. Shen, ”Non-texture inpaintings by curvature-driven diffusions”, *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436-449, 2001.

[10] S. Esedoglu, J. Shen, ”Digital Inpainting Based on the Mumford-Shah-Euler Image Model”, *European Journal of Applied Mathematics*, vol. 13, pp. 353-370, 2002.

[11] A. Efros, T. Leung, ”Texture Synthesis by Non-parametric Sampling”, *in Proc. 1999 IEEE International Conference on Computer Vision (ICCV)*, pp. 1033–1038, 1999.

[12] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, ”Simultaneous Structure and Texture Image Inpainting”, *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882-889, August 2003.

[13] I. Giakoumis, I. Pitas, ”Digital Restoration of Painting Cracks”, *in Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 269-272, 1998.

[14] F. Meyer, “Iterative image transforms for an automatic screening of cervical smears,” *J.Histoch.Cytochem.*, vol.27, pp. 128–135, 1979.

[15] I. Pitas, A.N. Venetsanopoulos, *Nonlinear Digital Filters, principles and applications*, Norwell: Kluwer Academic, 1990.

[16] A. G. Bors, I. Pitas, ”Median Radial Basis Function Neural Network”, *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1351-1364, November 1996

[17] S. Haykin, *Neural Networks, A Comprehensive Fountation*, 2nd Edition, New York: Prentice Hall, 1999.

[18] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, M. Gabbouj, ”Order Statistics Learning Vector Quantizer”, *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1048-1053, June 1996.

[19] G. Seber, *Multivariate Observations*, New York: John Wiley, 1986.

[20] P. Perona, J. Malik, “Scale-Space and Edge Detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.

[21] P. Hough, “A Method and Means for Recognizing Complex Patterns,” *U.S. Patent*, 3 069 654, 1962.

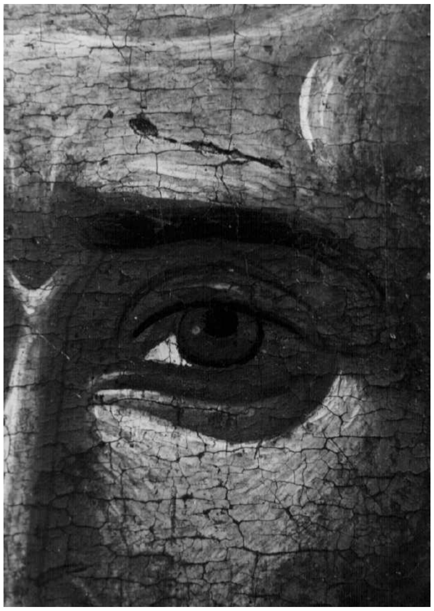

Fig. 1. Original painting.

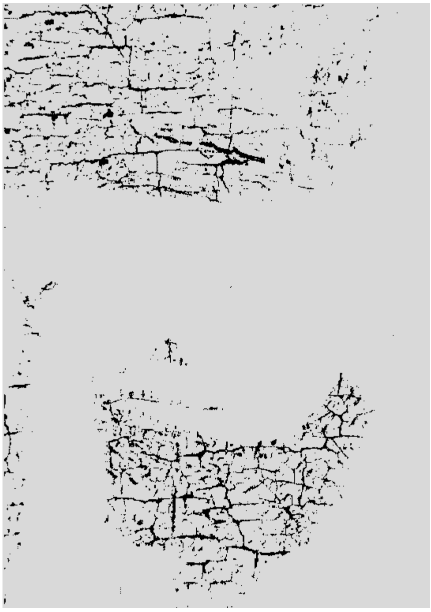

Fig. 2. Thresholded output of the top hat transform (threshold value: 23).

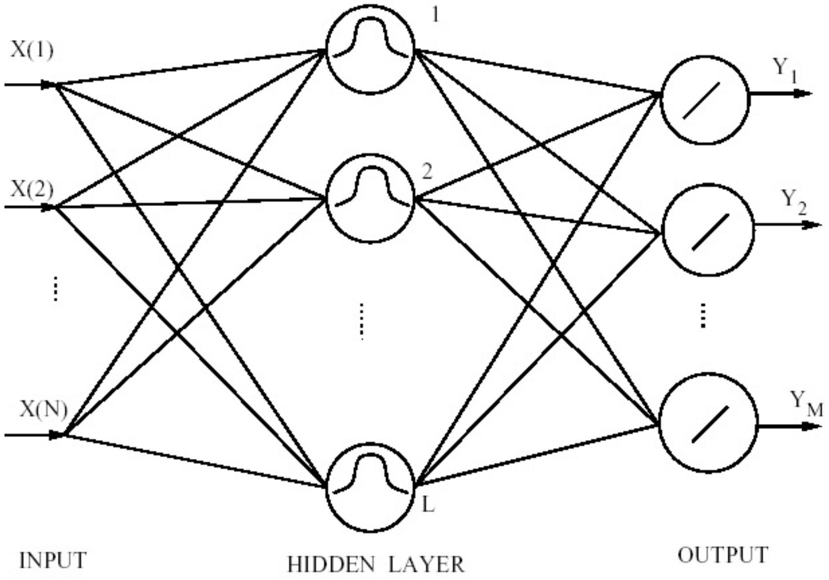

Fig. 3. Radial Basis Functions neural network architecture.

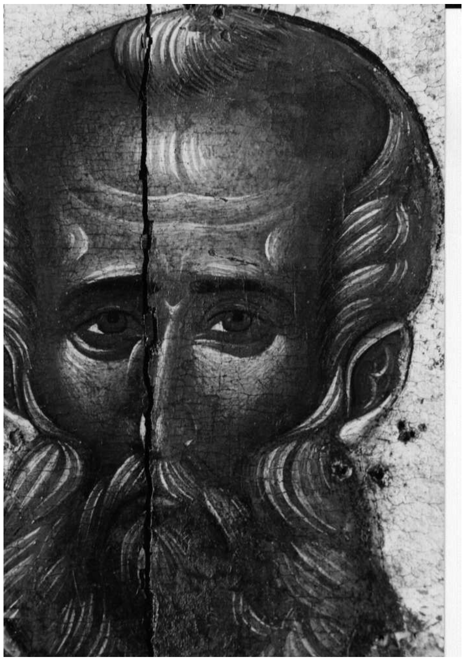

Fig. 4. Original painting.

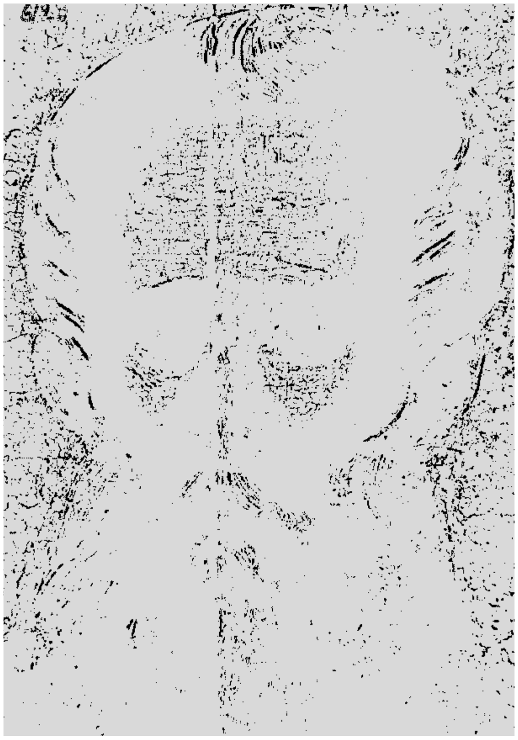

Fig. 5. Thresholded output of the top-hat transform (threshold value: 19). A number of brush strokes (hair, number in the upper left corner) have been misidentified as cracks.

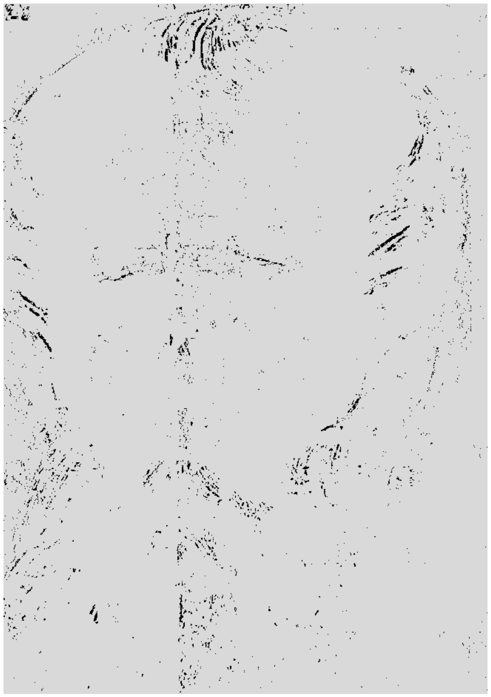

Fig. 6. The separated brush strokes after the application of the MRBF technique.

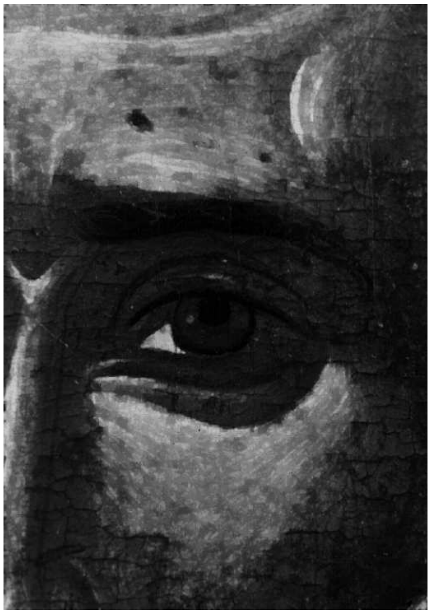

Fig. 7. Crack filling by using the modified trimmed mean filter (filter size 5×5).

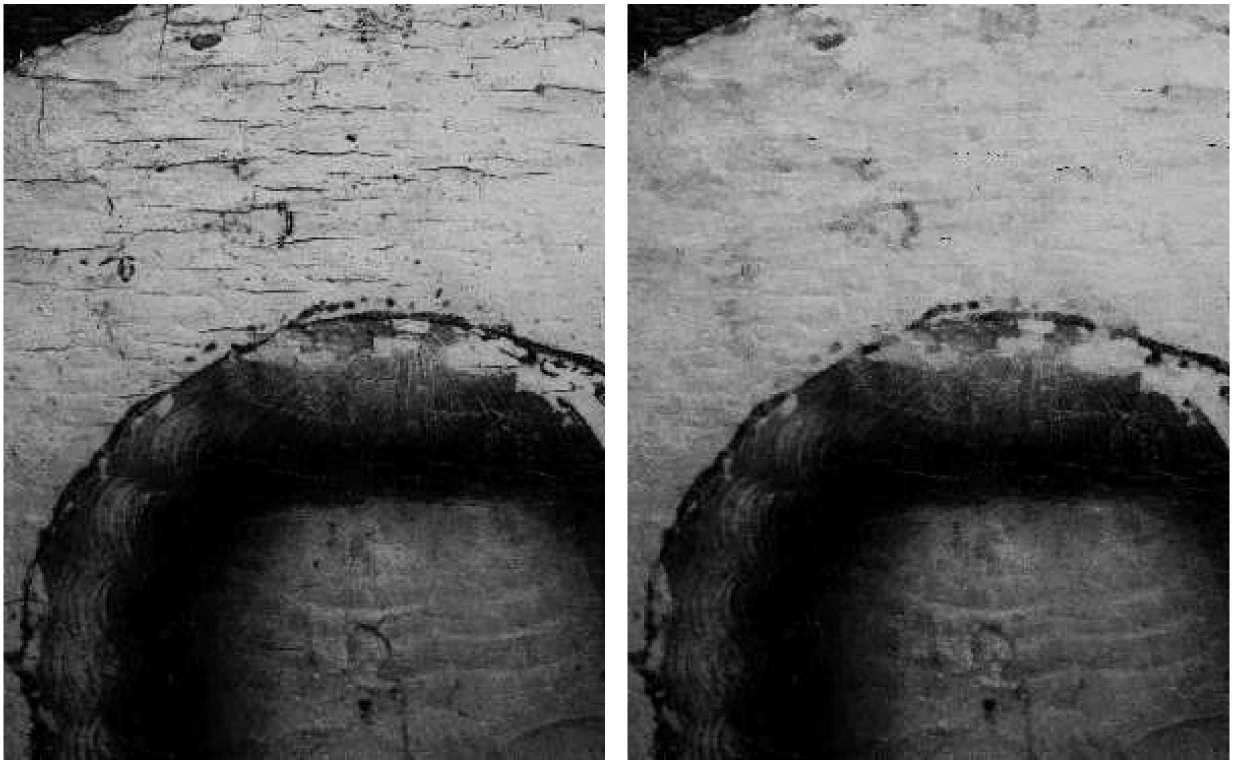

Fig. 8. (a) Original painting (detail). (b) Crack filling by using the modified trimmed mean filter (filter size 3×3).

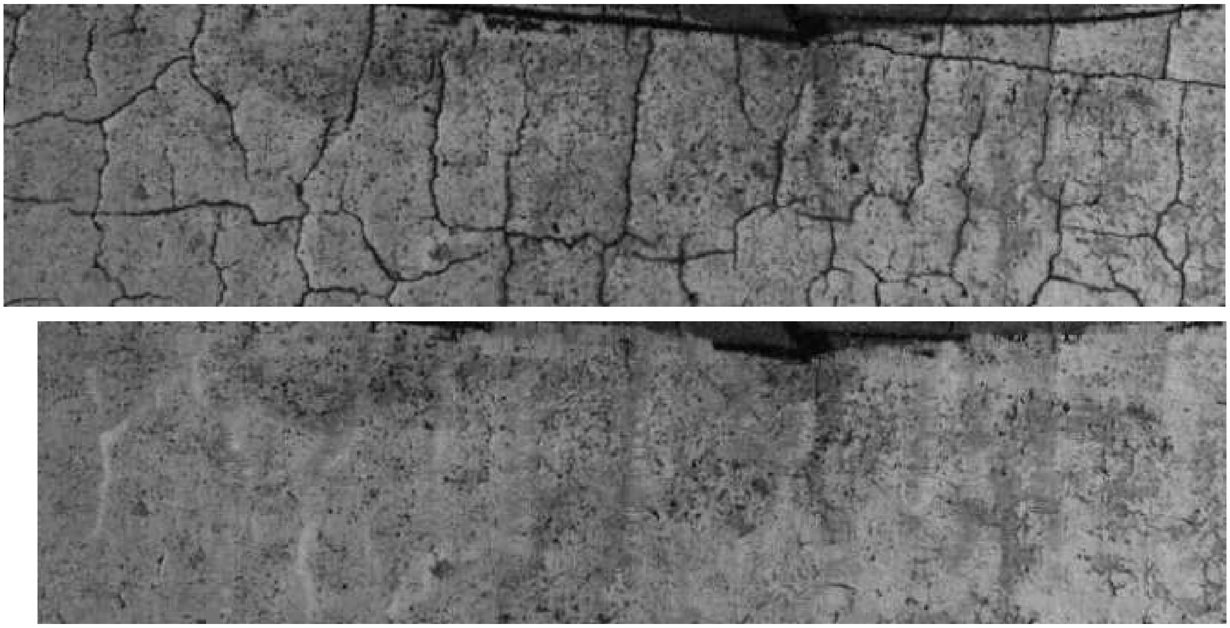

Fig. 9. (a) Original image with cracks (detail), (b) crack filling using the orientation-sensitive controlled anisotropic diffusion technique.

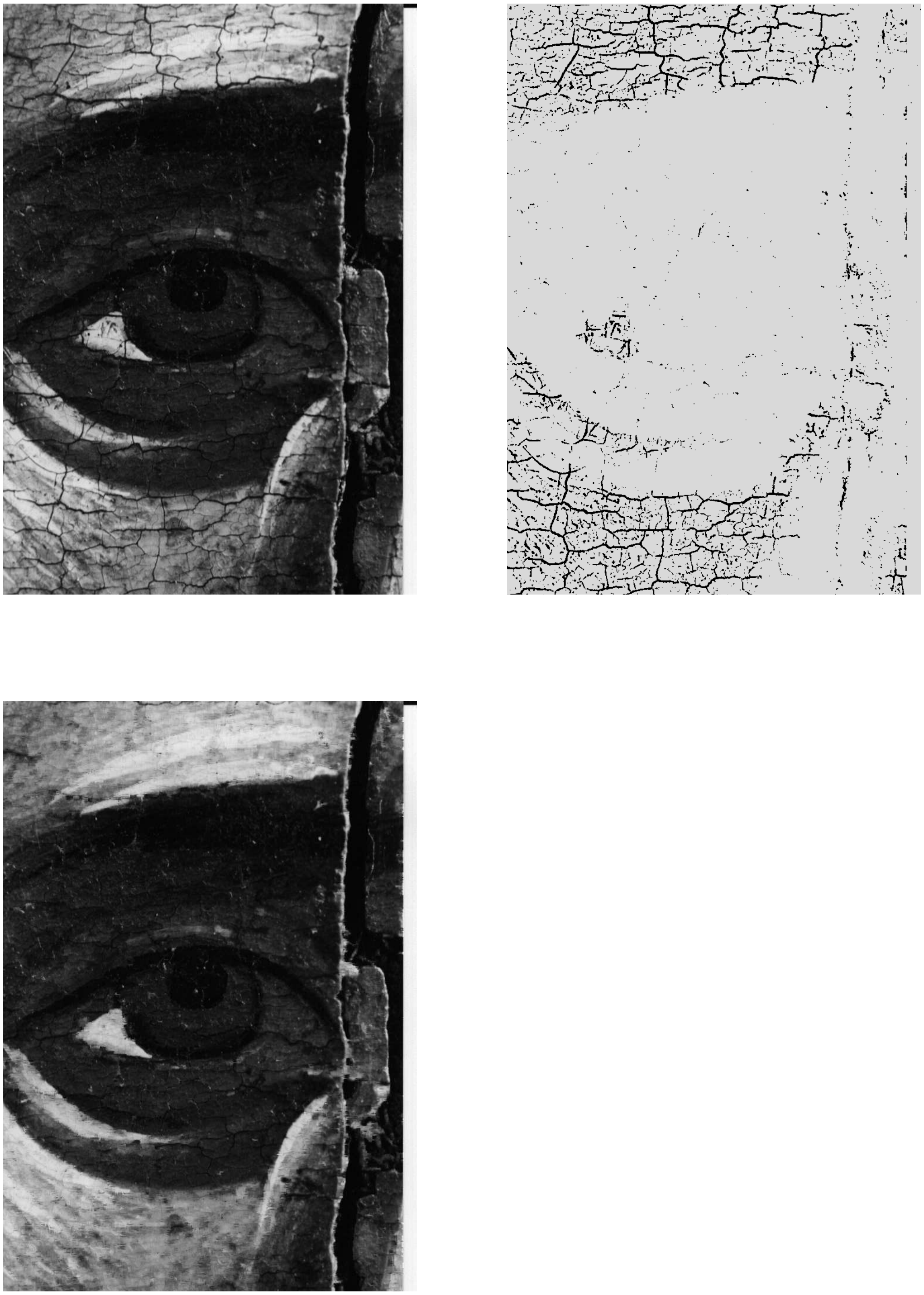

Fig. 10. (a) Original painting (detail), (b) crack image (threshold value: 17), (c) crack filling using the orientation-sensitive controlled anisotropic diffusion technique.

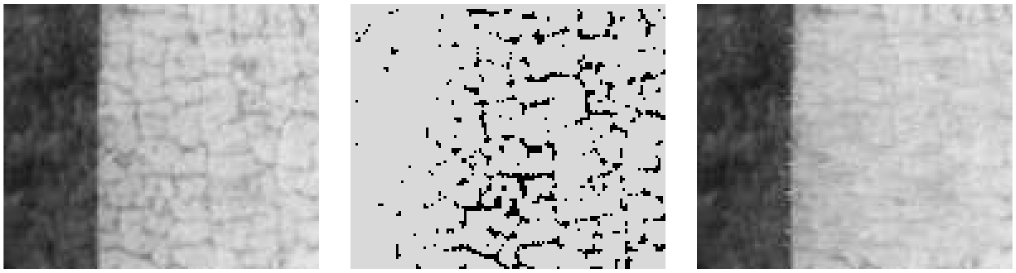

Fig. 11. (a) Original painting (detail), (b) crack image (threshold value: 21), (c) crack filling using the orientation-sensitive controlled anisotropic diffusion technique.

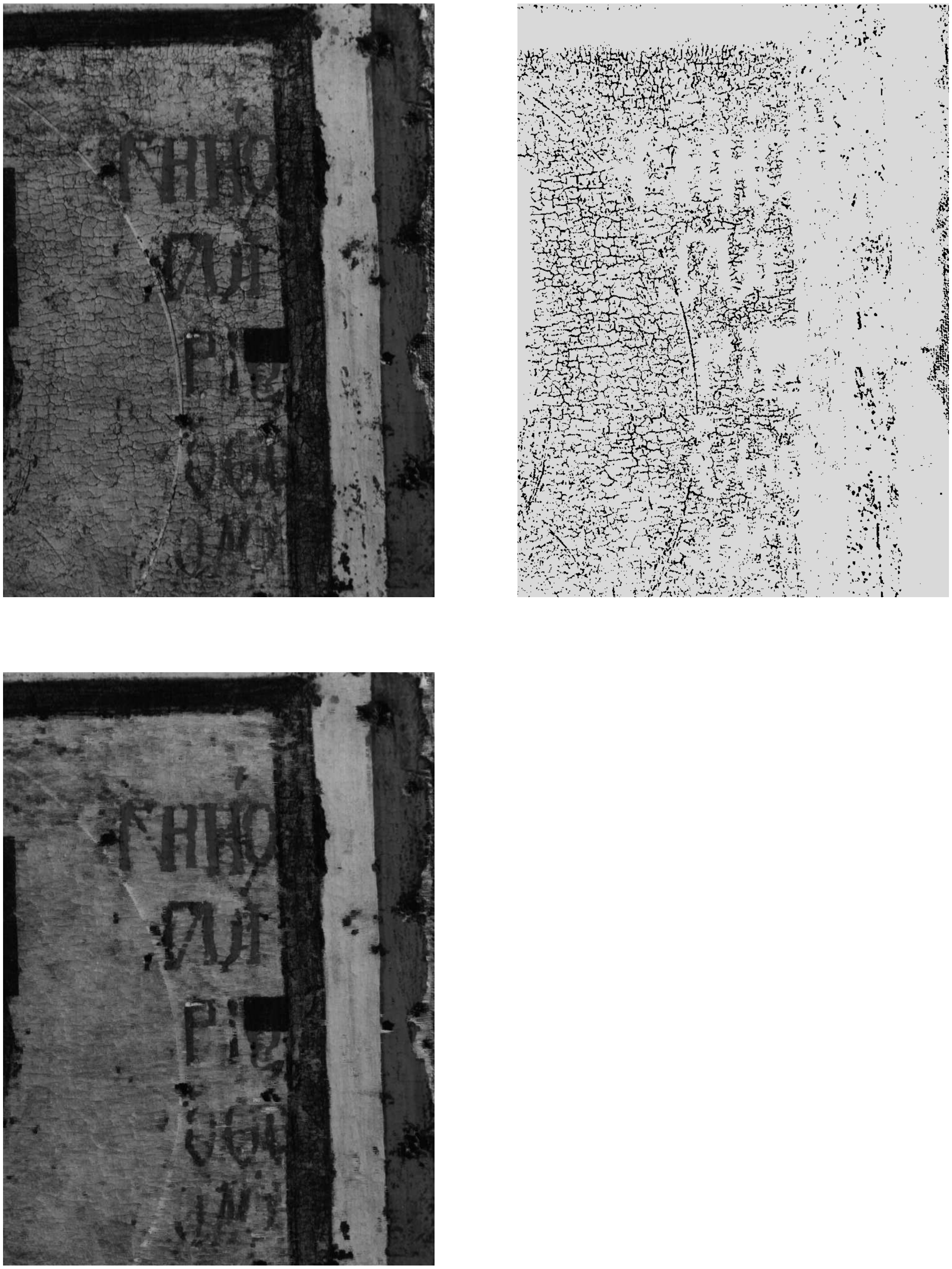

Fig. 12. (a) Original painting (detail), (b) crack image (threshold value: 23), (c) crack filling using the orientation-sensitive controlled anisotropic diffusion technique.

Ioannis Giakoumis received the Diploma of Informatics in 1997. He is currently with Intracom S.A.

PLACE PHOTO HERE

Nikos Nikolaidis received the Diploma of Electrical Engineering in 1991 and the PhD degree in Electrical Engineering in 1997, both from the Aristotle University of Thessaloniki, Greece. From 1992 to 1996 he served as teaching assistant in the Departments of Electrical Engineering and Informatics at the same University. From 1998 to 2002 he was postdoctoral researcher and teaching assistant at the Department of Informatics, Aristotle University of Thessaloniki. He is currently a Lecturer in the same Department. Dr. Nikolaidis is the co-author of the book ”3-D Image Processing Algorithms” (J. Wiley, 2000). He has co-authored 2 book chapters, 18 journal papers and 46 conference papers. His research interests include computer graphics, image and video processing and analysis, copyright protection of multimedia and 3-D image processing. Dr. Nikolaidis has been a scholar of the State Scholarship Foundation of Greece.

PLACE PHOTO HERE

Ioannis Pitas received the Diploma of Electrical Engineering in 1980 and the PhD degree in Electrical Engineering in 1985 both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate at the University of Toronto, Canada, University of Erlangen-Nuernberg, Germany, Tampere University of Technology, Finland, as Visiting Assistant Professor at the University of Toronto and as Visiting Professor at the University of British Columbia, Vancouver, Canada. He was lecturer in short courses for continuing education. He has published 140 journal papers, 350 conference papers and contributed in 18 books in his areas of interest. He is the co-author of the books ”Nonlinear Digital Filters: Principles and Applications” (Kluwer, 1990), ”3-D Image Processing Algorithms” (J. Wiley, 2000), Nonlinear Model-Based Image/Video Processing and Analysis (J. Wiley, 2001) and author of ”Digital Image Processing Algorithms and Applications” (J. Wiley, 2000). He is the editor of the book ”Parallel Algorithms and Architectures for Digital Image Processing, Computer Vision and Neural Networks” (Wiley, 1993). He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as Associate Editor of the IEEE Transactions on Circuits and Systems, IEEE Transactions on Neural Networks, IEEE Transactions on Image Processing, EURASIP Journal on Applied Signal Processing and co-editor of Multidimensional Systems and Signal Processing. He was general chair of the 1995 IEEE Workshop on Nonlinear Signal and Image Processing (NSIP95), technical chair of the 1998 European Signal Processing Conference and general chair of IEEE ICIP 2001. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.

LIST OF FIGURES